

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-049497

(43)Date of publication of application : 15.02.2002

(51)Int.Cl.

G06F 9/46

(21)Application number : 2000-237269

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 04.08.2000

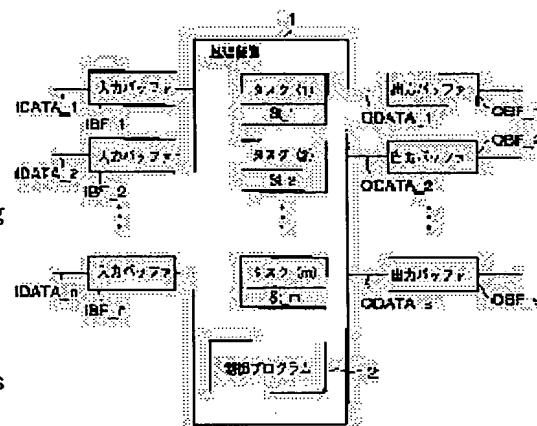
(72)Inventor : TAKAHASHI KATSUMI

(54) DATA PROCESSING METHOD AND PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a multitask data processing method capable of performing multitask processing corresponding to plural data to be processed that the current states of I/O systems of plural processing components are changed every moment and suited to streaming, and to provide a processor using the method.

SOLUTION: In the case of switching tasks to be processed in order to execute plural different processing components to be applied to plural data to be processed by multitask processing, the current states of I/O systems of plural processing components for respective tasks are evaluated, the priority of respective I/O systems is calculated and a task to be processed next is determined on the basis of the priority.



LEGAL STATUS

[Date of request for examination]

24.07.2002

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-49497

(P2002-49497A)

(43)公開日 平成14年2月15日(2002.2.15)

(51)Int.Cl.⁷

G 0 6 F 9/46

識別記号

3 4 0

F I

C 0 6 F 9/46

テーム(参考)

3 4 0 B 5 B 0 9 8

審査請求 未請求 請求項の数4 O L (全 5 頁)

(21)出願番号 特願2000-237269(P2000-237269)

(22)出願日 平成12年8月4日(2000.8.4)

(71)出願人 000003078

株式会社東芝

東京都港区芝浦一丁目1番1号

(72)発明者 高橋 克己

神奈川県川崎市幸区小向東芝町1番地 株

式会社東芝研究開発センター内

(74)代理人 100058479

弁理士 鈴江 武彦 (外6名)

Fターム(参考) 5B098 AA08 CC01 GA04 GC03 GD02

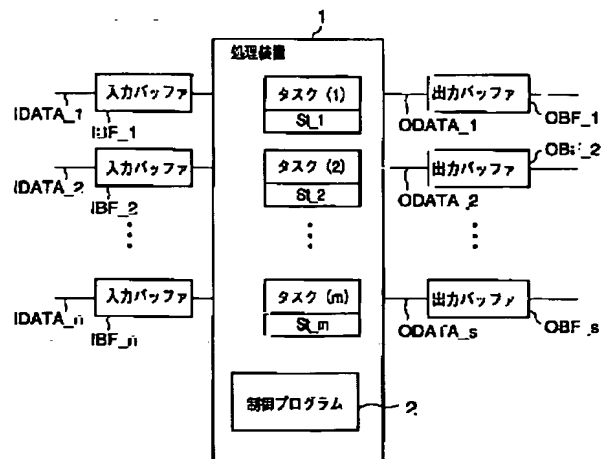
GD17

(54)【発明の名称】 データ処理方法および処理装置

(57)【要約】

【課題】複数の処理の入出力系統の現状が時々刻々と変化するような複数の処理対象データに対するマルチタスク処理を可能にし、ストリーミングに適したマルチタスクデータ処理方法およびそれを用いた処理装置を提供する。

【解決手段】複数の処理対象データに対し複数の異なる処理をマルチタスク処理にて実行するため、処理対象のタスクを切り換えるにあたって、各タスクについて前記複数の処理の入出力系統の現状に対する評価を行ってそれぞれの優先度を算出し、この優先度に基づき次の処理対象のタスクを決定する。



【特許請求の範囲】

【請求項1】 複数の処理対象データに対し複数の異なる処理をマルチタスク処理にて実行するためのデータ処理方法において、

処理対象のタスクを切り換えるにあたって、各タスクについて前記複数の処理の入出力系統の現状に対する評価を行ってそれぞれの優先度を算出し、この優先度に基づき次の処理対象のタスクを決定することを特徴とするデータ処理方法。

【請求項2】 複数の処理対象データに対し複数の異なる処理をマルチタスク処理にて実行するためのデータ処理方法において、

処理対象のタスクを切り換えるにあたって、各タスクについて前記複数の処理の入出力系統の現状に対する評価を行うための関数を用いて、それぞれのタスクの優先度を算出し、この優先度に基づき次の処理対象のタスクを決定することを特徴とするデータ処理方法。

【請求項3】 複数の処理対象データに対し複数の異なる処理をマルチタスク処理にて実行する処理装置において、

処理対象のタスクを切り換えるにあたって、各タスクについて前記複数の処理の入出力系統の現状に対する評価を行ってそれぞれの優先度を算出する算出手段を具備し、この算出された優先度に基づき次の処理対象のタスクを決定することを特徴とする処理装置。

【請求項4】 複数の処理対象データに対し複数の異なる処理をマルチタスク処理にて実行する処理装置において、

処理対象のタスクを切り換えるにあたって、各タスクについて前記複数の処理の入出力系統の現状に対する評価を行うための関数を用いて、それぞれのタスクの優先度を算出する算出手段を具備し、この算出された優先度に基づき次の処理対象のタスクを決定することを特徴とする処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、例えば、インターネットやデジタル放送などで音楽や映像などのデータを受信しながらリアルタイムに再生する（ストリーミング）際のマルチタスクによるデータ処理方法およびそれを用いたデータ処理装置に関する。

【0002】

【従来の技術】複数のタスクを単一の処理装置（プロセッサ）で処理する場合、時間分割して処理することが行われている。ある時刻には1個のタスクが実行されており、このタスクの実行が中断されると、制御プログラムに制御が移され、制御プログラムは次に実行するタスクを決定する。従来の方式では、制御プログラムは、実行可能なタスクの中で、各タスクの優先順位を示す、あらかじめ数値化された値の大小に基づいて、次に実行する

タスクを決定する。詳細な決定アルゴリズムは幾通りがあるが、代表的なものは次の通りである。実行可能なタスクの中で、最も優先度の高いものが1個ある場合は、それが選択される。最も優先度の高いものが2個以上ある場合には、それらがサイクリックに順番に実行されるように決定される。

【0003】優先順位を示す数値は、タスクの実行中に固定であるものや変化するものがある。変化するものは、タスク自身が値を変更するのが普通の方法である。つまりプログラムの実行位置に応じて、高い優先度が必要になった場合には優先度を上げ、低い優先度でも良くなった場合には優先度を下げるのである。

【0004】ところが、インターネットやデジタル放送などで音楽や映像などのデータを受信しながらリアルタイムに再生する（ストリーミング）際には、連続的に入力されてくる音声データなどストリーミング・データを処理するプログラムにおいては、未処理のデータが格納されるバッファの占有量など、プログラムの実行位置でなく、外的要因に応じて、優先度を変化させる必要がある。すなわち、未処理の入力データが少ないときは処理を急がないが、未処理の入力データが多いときは処理を急がないと入力データがオーバーフローしてしまい、データが欠落してしまう。あるいは、一定の遅延以内で処理を行わないと、出力データに時間的切れ目が生じてしまい、音声データならば音がとぎれるという問題が発生する。

【0005】

【発明が解決しようとする課題】従来のマルチタスク処理は、固定の優先順位や、プログラム自身が設定する優先度に基づいて処理対象のタスクの切り換えを制御するものであり、複数の処理の入出力系統の現状（入出力バッファ内のデータ量）を評価してタスクの優先度を定める必要のあるストリーミングを行う際には適していないという問題点があった。

【0006】そこで、本発明は、複数の処理の入出力系統の現状が時々刻々と変化するような複数の処理対象データに対するマルチタスク処理を可能にし、ストリーミングに適したマルチタスクデータ処理方法およびそれを用いた処理装置を提供することを目的とする。

【0007】

【課題を解決するための手段】本発明は、複数の処理対象データに対し複数の異なる処理をマルチタスク処理にて実行するためのデータ処理方法であって、処理対象のタスクを切り換えるにあたって、各タスクについて前記複数の処理の入出力系統の現状に対する評価を行ってそれぞれの優先度を算出し、この優先度に基づき次の処理対象のタスクを決定することにより、複数の処理の入出力系統の現状が時々刻々と変化するような複数の処理対象データに対するデータ処理をマルチタスク処理にて実行することが可能となる。

【0008】好ましくは、各タスクについて前記複数の処理の入出力系統の現状に対する評価を行うための関数を用いて、それぞれのタスクの優先度を算出する。

【0009】本発明は、複数の処理対象データに対し複数の異なる処理をマルチタスク処理にて実行する処理装置において、処理対象のタスクを切り換えるにあたって、各タスクについて前記複数の処理の入出力系統の現状に対する評価を行ってそれぞれの優先度を算出する算出手段を具備し、この算出された優先度に基づき次の処理対象のタスクを決定することにより、複数の処理の入出力系統の現状が時々刻々と変化するような複数の処理対象データに対するデータ処理をマルチタスク処理にて実行することが可能となる。

【0010】好ましくは、各タスクについて前記複数の処理の入出力系統の現状に対する評価を行うための関数を用いて、それぞれのタスクの優先度を算出する。

【0011】

【発明の実施の形態】以下、本発明の実施形態について図面を参照して説明する。

【0012】図1は、本実施形態に係るデータ処理装置（以下、簡単に処理装置と呼ぶ）1の要部の構成を概念的に示したものである。処理対象のデータが複数（ここでは、例えばn系列）あり（入力データIDATA_1～IDATA_n）、それらはそれぞれ入力バッファIBF_1～入力バッファIBF_nに入力される。これら入力データは、例えば、複数チャネルのオーディオ信号やビデオ信号などである。これらのデータは、マクロ的にはほぼ一定のレートで入力されることが多いが、ミクロ的には入力バッファに転送する側の性質に依存するものであり、バースト的に入力される場合もありうる。

【0013】図1における処理装置1において、各入力データに対する処理が遅れると、入力バッファがオーバーフローしてしまうため、それが起こらないように処理しなければならない。

【0014】処理装置1は、ハードウェア的には単一のプロセッサで構成されており、この上で複数の（ここでは、例えばm個）のタスク（タスク(1)～タスク(m)）が動作するものとする。

【0015】タスク(1)～タスク(m)は、入力バッファIBF_1～IBF_nに格納されているデータを逐次処理していくための処理プログラムである。

【0016】入力バッファIBF_1～IBF_nに格納されている入力データIDATA_1～IDATA_nは、タスク(1)～タスク(m)で処理されて、複数の（ここでは、例えばs系列）の出力データODATA_1～ODATA_sとして出力され、それぞれ出力バッファOBF_1～OBF_sに格納される。なお、出力バッファがそのまま入力バッファに対応することもあるが、ここでは、説明の簡単のため、それぞれ別個に設けられているものとする。なお、 $m=n$ およびまたは $n=s$

sであってもよい。

【0017】図1の処理装置1は単一のプロセッサで構成されているので、同時には1個のタスクしか実行されず、m個のタスクは時間分割されて実行されることになる。なお、ここでは、単一のプロセッサで構成された処理装置の場合を例にとり説明するが、一般的には、mより少ない複数のプロセッサで構成された処理装置でも同様である。

【0018】各タスクへの時間割り当てを決めるのは、制御プログラム2である。タスクの時間割り当ては次のようにして行われる。ある時刻にタスク(k) ($1 \leq k \leq m$ の整数) が実行されているとする。

【0019】タスク(k)はある条件に従って処理が中断される。この条件は、従来の一般的なマルチタスク制御方式においては、タスク(k)の実行が始まってから一定時間が経過したときである。すなわち、時間を固定長のタイム・スロットに分割し、各タイム・スロット毎にタスクを割り当てる方式である。

【0020】本発明においても、このような方式を用いることも可能である。また、この方式を用いない場合としては、タスク(k)が自ら実行を中断する方式が考えられる。後者の方式による利点は、自らの都合で中断を行えるため、処理上区切りがよいところで中断することにより、タスク切り替え時の時間が短くなり、効率がよいことである。すなわち逆に言えば、前者のように強制的に中断される方式では、後で再開するときに必要な中間データが大きくなり、中断時にそれらを保存したり、再開時にそれらを復元したりするのに、より多くの時間を費やすことになる。後者の方式における処理上の区切りとは、例えば、入力データが圧縮符号化されたオーディオ信号の場合には、フレームの区切りとすると都合がよい。

【0021】本実施形態では、前者の予め定められた時間経過したときタスクの処理を中断させてタスクを切り換える方式であっても、後者のタスク自身が例えばフレームの区切りをきっかけに処理を中断してときタスクを切り換える方式であっても適用可能である。

【0022】以下、図2に示すフローチャートを参照して、制御プログラム2のマルチタスク制御処理動作について説明する。

【0023】タスク(k)が処理装置のプロセッサにより実行され、その実行が中断されると、制御プログラム2に処理が移される。制御プログラム2は、OS（オペレーションシステム）が存在するシステムの場合には、通常、OSの一部である。制御プログラム2は、ある条件により次に実行するタスクを決定するものである。

【0024】従来の制御プログラムにおいては、各タスクに優先度を示す数値が設定されており、実行可能なタスクの中で、それらの数値の最も大きいものが次に実行されるタスクとして選択される。ここで、実行可能と

は、図1のような処理系の場合には、各タスクに対応する処理の入出力系統において、その入力バッファが空でなく、かつ対応する出力バッファがフルでないことである。

【0025】しかしながら、図1に示す処理系では、各タスクにあらかじめ優先度をつけておくことや、タスク実行時に自分自身の優先度を設定しておくことは望ましくない。なぜなら、ここでの優先度のあるべき姿は、入力バッファがオーバーフローしないことや、出力バッファが空にならないことであり、これらの条件は当該タスクが実行していない場合にも時々刻々と変化するからである。すなわち優先度と表す数値が時々刻々と変化するようになっている必要がある。

【0026】本発明では、時々刻々と変化する優先度を実現するために、各タスク(1)～タスク(m)のそれぞれにステータス関数 $st_1 \sim st_m$ を対応付け、制御プログラム2が次に実行すべきタスクを選択する時点(タスクの切り換え時)において、すなわち、今まで実行していたタスクが中断した後に、各タスク(1)～タスク(m)のそれぞれに対応したステータス関数 $st_1 \sim st_m$ を呼び(ステップS1)、各ステータス関数により、例えば、後述する算出方法にて優先度を算出した結果を取得する(ステップS2)。

【0027】制御プログラムは、各ステータス関数から得られた値、すなわち、各タスクの優先度から、次に実行すべきタスクとして、最も優先度の高いタスクを選択する。ここでは、その選択されたタスクをタスク(k)とする(ステップS3)。制御プログラムは、タスクの切り換え時に、プロセッサに、この選択されたタスク(k)を次の処理対象として実行させる。

【0028】各ステータス関数 $st_1 \sim st_m$ は、それらに対応したタスクの入出力系統の現状を評価し、それに応じた優先度を返すように作られている。また各タスクが実行可能か実行不可能かの情報も返すようにすることも可能なようになっている。ここで実行不可能とは、例えば入力バッファ占有量が空か、出力バッファ占有量がフルの場合である。

【0029】制御プログラム2は、ステップS3において、これらすべてのステータス関数と呼んだ後、実行可能なタスクの内でも優先度が高いものを、次に実行するタスクとして選択する。

【0030】次に、ステータス関数の優先度の算出方法について、ここでは2例を挙げて説明する。なお、ステータス関数の優先度の算出方法は、ここに挙げるもののみに限るものではない。

【0031】第1の算出方法としては、各ステータス関数は、それに対応した入力バッファの占有量を調べ、その占有量ある規則に従って優先度に変換する。ある規則とは、例えば、占有量とあらかじめ予想される入力データのレートから、オーバーフローするまでの時間を推

測し、その時間が短いほど優先度が高くなり、その時間が長いほど優先度が低くなるようなテーブルにより変換する。

【0032】第2の算出方法としては、各ステータス関数は、それに対応したタスクが処理を完了しているデータのタイム・スタンプを調べ、その時刻と実際の時刻との差を優先度に変換して返す。この場合は、入力データ $IDATA_1 \sim IDATA_n$ がすべてタイム・スタンプを持ったデータであることが条件である。また、処理装置1には現在時刻を表す時計またはカウンタが具備されているものとする。この例では、実時間に対する遅延が最も大きいものに最も高い優先度が与えられる。

【0033】以上説明したように、上記実施形態によれば、各タスクにその対応する処理の入出力系統の現状に対する評価を行うためのステータス関数を対応づけ、制御プログラム2は、処理対象のタスクの切り換えるにあたって、各タスクのステータス関数から返ってきた値をタスクの優先度として用いて、次の処理対象のタスクを決定することにより、複数の処理の入出力系統の現状が時々刻々と変化するような複数の処理対象データに対するデータ処理をマルチタスク処理にて実行することが可能となり、例えば、ストリーミングの際に、処理が所定の時刻に間に合うように行われ、入力バッファがオーバーフローしたり、出力がとぎれたりすることがなくなる。

【0034】なお、本発明は、上記実施形態に限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で種々に変形することが可能である。さらに、上記実施形態には種々の段階の発明は含まれており、開示される複数の構成要件における適宜な組み合わせにより、種々の発明が抽出され得る。例えば、実施形態に示される全構成要件から幾つかの構成要件が削除されても、発明が解決しようとする課題の欄で述べた課題(の少なくとも1つ)が解決でき、発明の効果の欄で述べられている効果(の少なくとも1つ)が得られる場合には、この構成要件が削除された構成が発明として抽出され得る。

【0035】

【発明の効果】以上説明したように、本発明によれば、複数の処理の入出力系統の現状が時々刻々と変化するような複数の処理対象データに対するデータ処理をマルチタスク処理にて実行することが可能となり、ストリーミングに適したマルチタスクデータ処理を実現できる。

【図面の簡単な説明】

【図1】本発明の実施形態に係るデータ処理装置の要部の構成を概念的に示した図。

【図2】制御プログラムのマルチタスク制御処理動作を説明するためのフローチャート。

【符号の説明】

$IDATA_1 \sim IDATA_n$ …入力データ

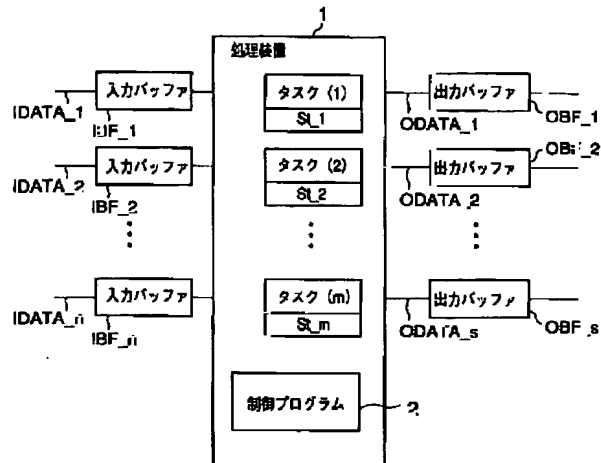
$IBF_1 \sim IBF_n$ …入力バッファ

St_1~St_m…ステータス関数

OBF_1~OBF_s…出力バッファ

ODATA_1~ODATA_s…出力データ

【図1】



【図2】

